US 20100333105A1

(19) United States
(12) Patent Application Publication
Horvitz et al.
(10) Pub. No.: US 2010/0333105 A1
(43) Pub. Date: Dec. 30, 2010

(54) PRECOMPUTATION FOR DATA CENTER LOAD BALANCING

(75) Inventors: Eric J. Horvitz, Kirkland, WA (US); Christian L. Belady, Mercer Island, WA (US)

Correspondence Address:
LEE & HAYES, PLLC
601 W. RIVERSIDE AVENUE, SUITE 1400
SPOKANE, WA 99201 (US)

(73) Assignee: MICROSOFT CORPORATION, Redmond, WA (US)

(21) Appl. No.: 12/492,385

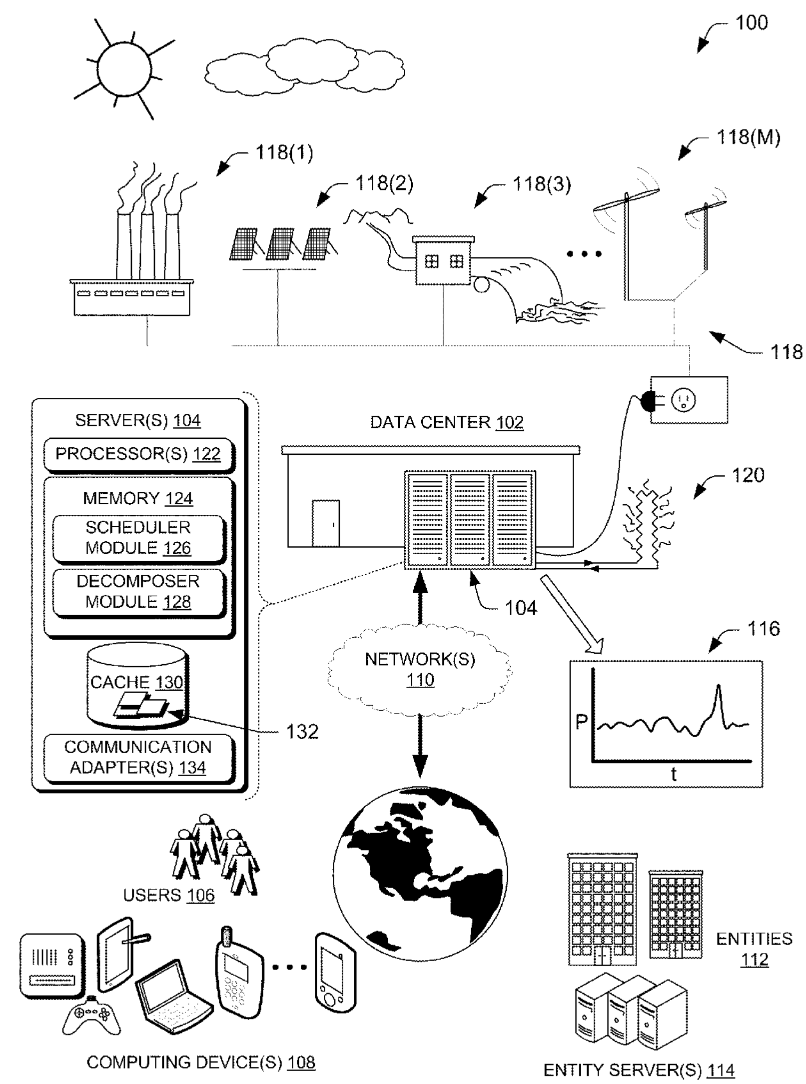
(22) Filed: Jun. 26, 2009

Publication Classification

(51) Int. Cl.
G06F 9/46 (2006.01)
G06F 12/00 (2006.01)
(52) U.S. Cl. 718/105; 707/759; 707/769; 707/713; 707/609

(57) ABSTRACT

Pre-computing a portion of forecasted workloads may enable load-balancing of data center workload, which may ultimately reduce capital and operational costs associated with data centers. Computing tasks performed by the data centers may be analyzed to identify computing tasks that are eligible for pre-computing, and may be performed prior to an actual data request from a user or entity. In some aspects, the pre-computing tasks may be performed during a low-volume workload period prior to a high-volume workload period to reduce peaks that typically occur in data center workloads that do not utilize pre-computation. Statistical modeling methods can be used to make predictions about the tasks that can be expected to maximally contribute to bottlenecks at data centers and to guide the speculative computing.



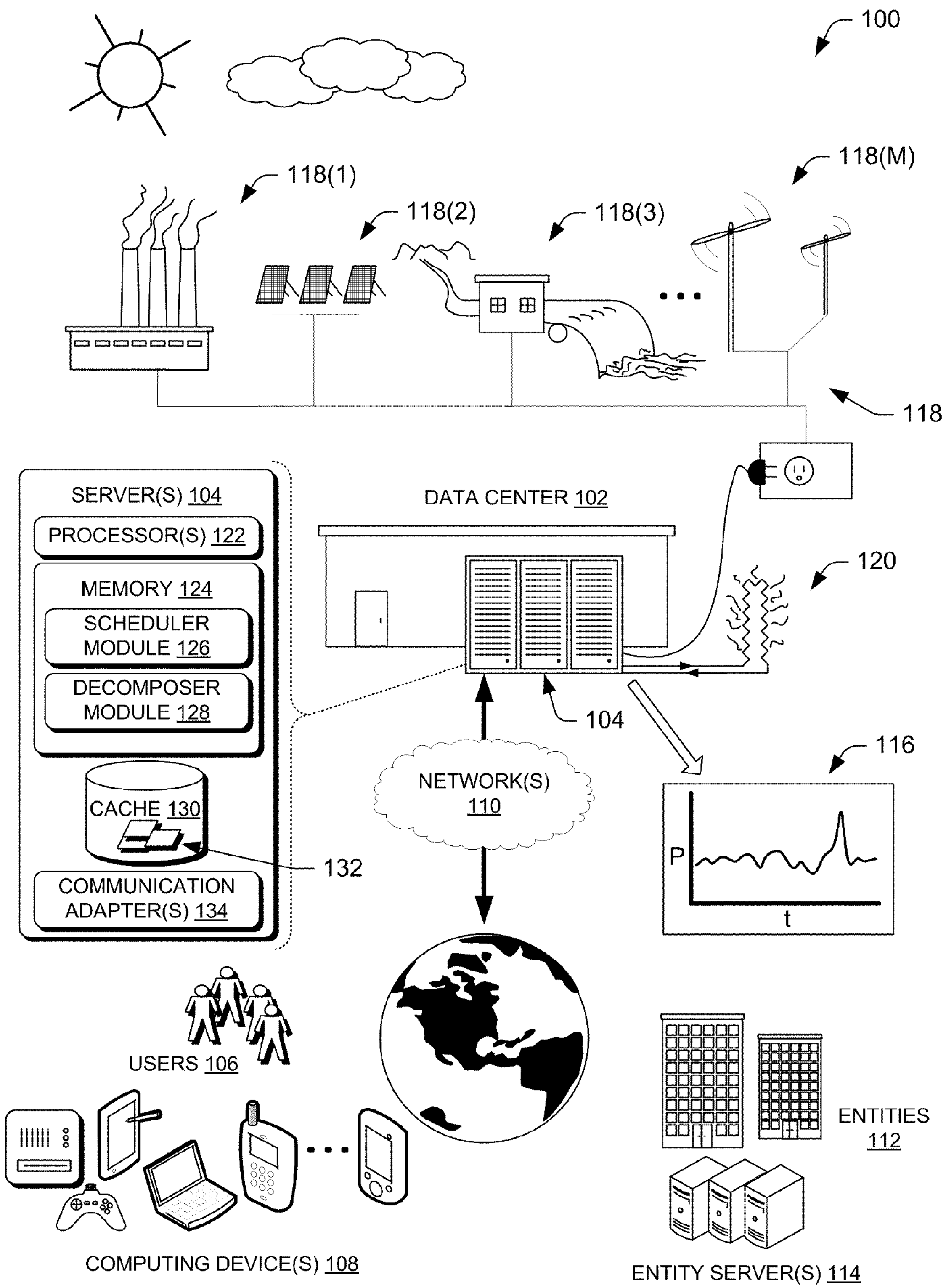

100
118(1)
118(2)
118(3)
118(M)
118
SERVER(S) 104
PROCESSOR(S) 122
MEMORY 124
SCHEDULER MODULE 126
DECOMPOSER MODULE 128
CACHE 130
COMMUNICATION ADAPTER(S) 134
132
DATA CENTER 102
120
104
NETWORK(S) 110
116
P
t
USERS 106
COMPUTING DEVICE(S) 108
ENTITIES 112
ENTITY SERVER(S) 114

FIG. 1

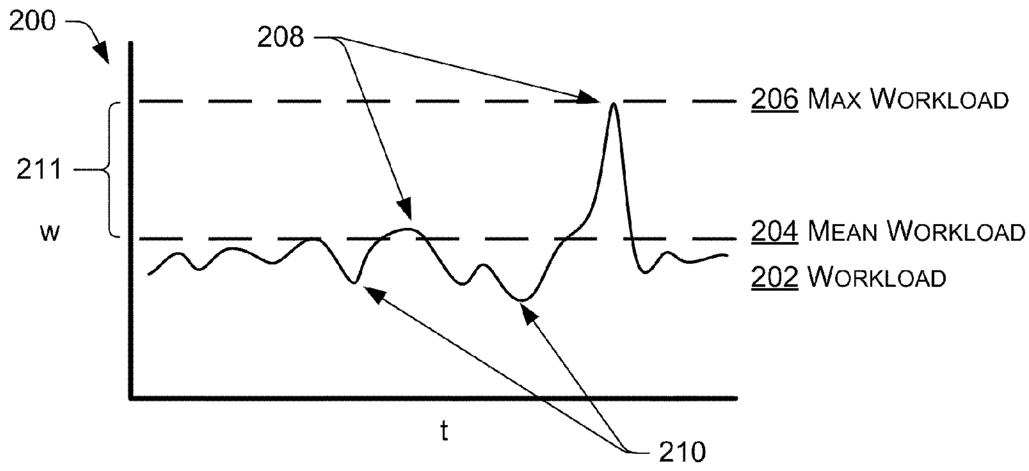


FIG. 2A

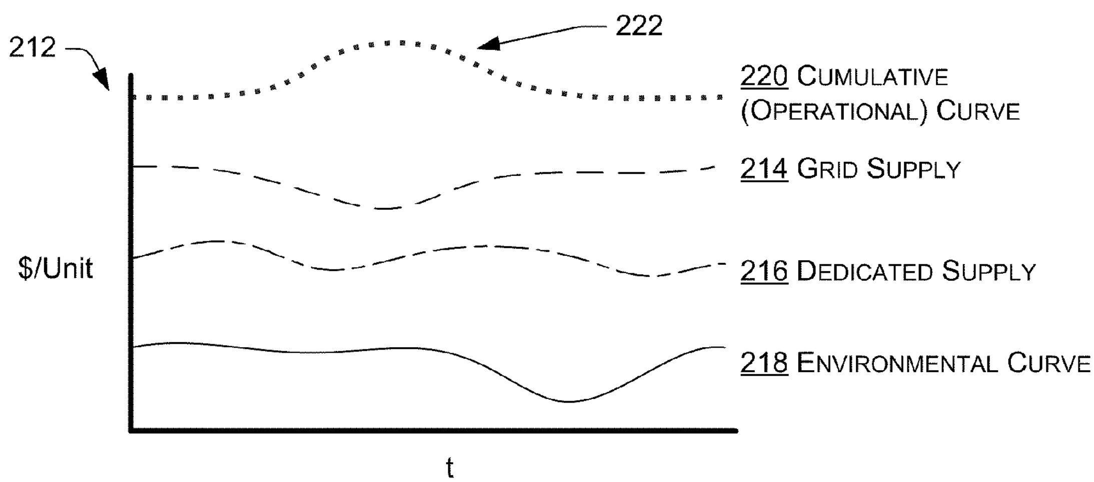


FIG. 2B

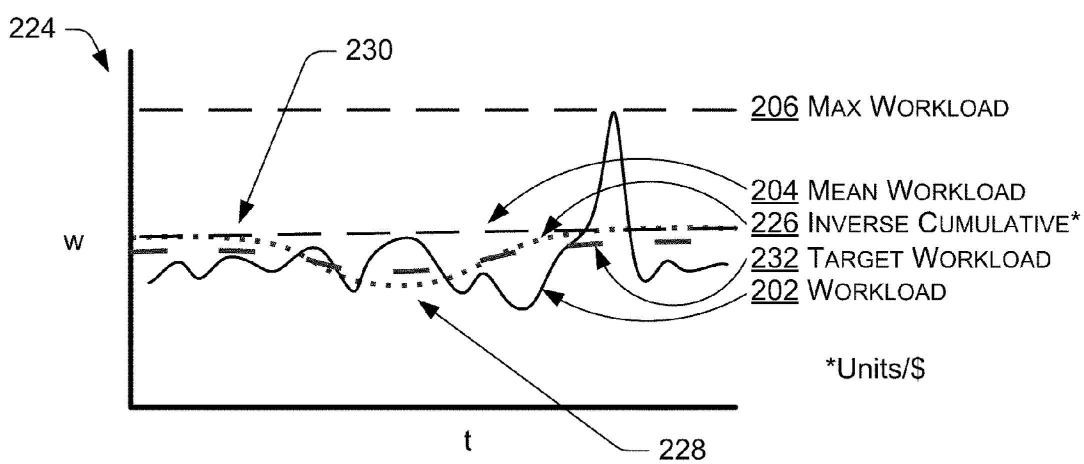


FIG. 2C

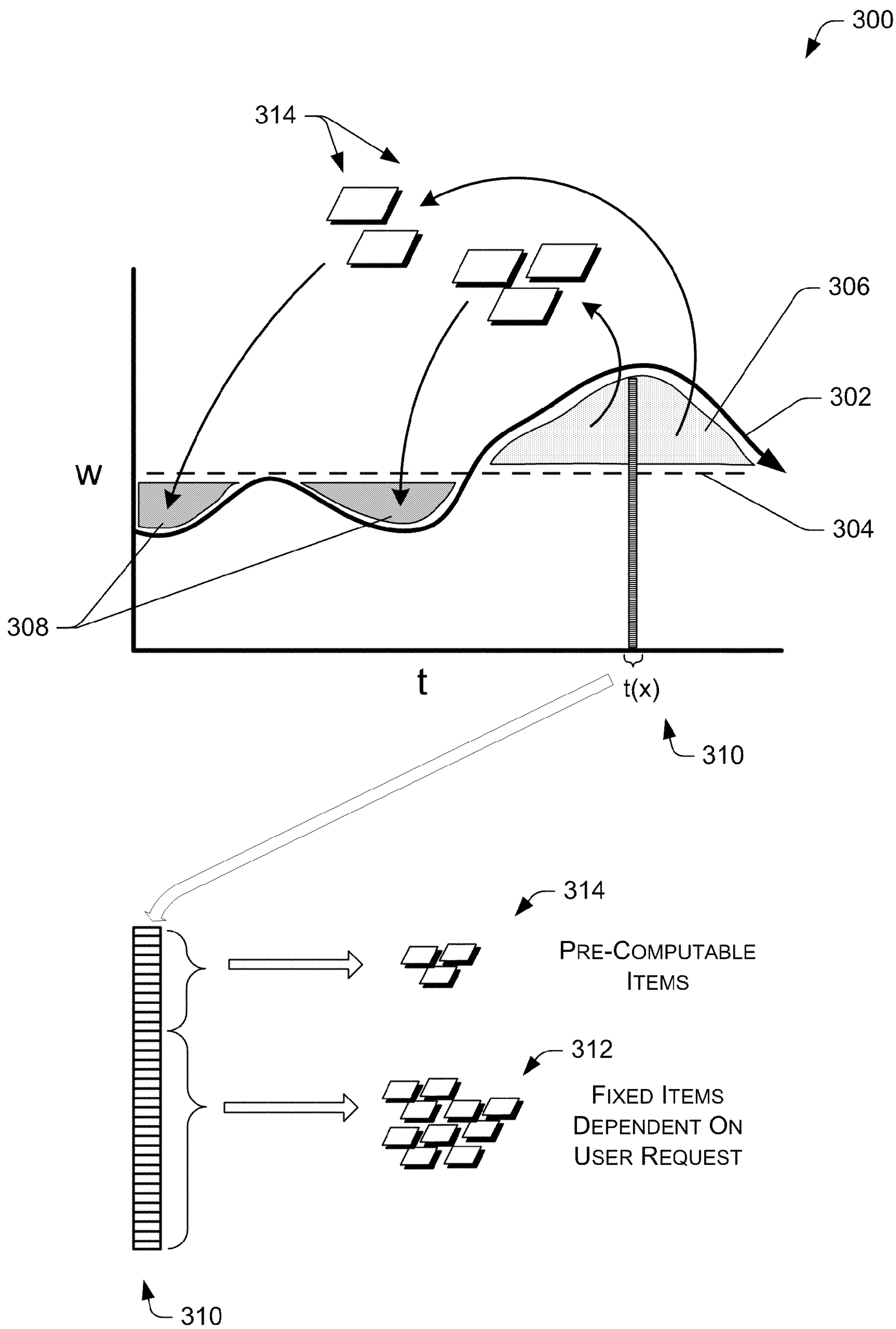

300
314
306
302
W
304
308
t
t(x)
310
314
PRE-COMPUTABLE ITEMS
312
FIXED ITEMS DEPENDENT ON USER REQUEST
310

FIG. 3

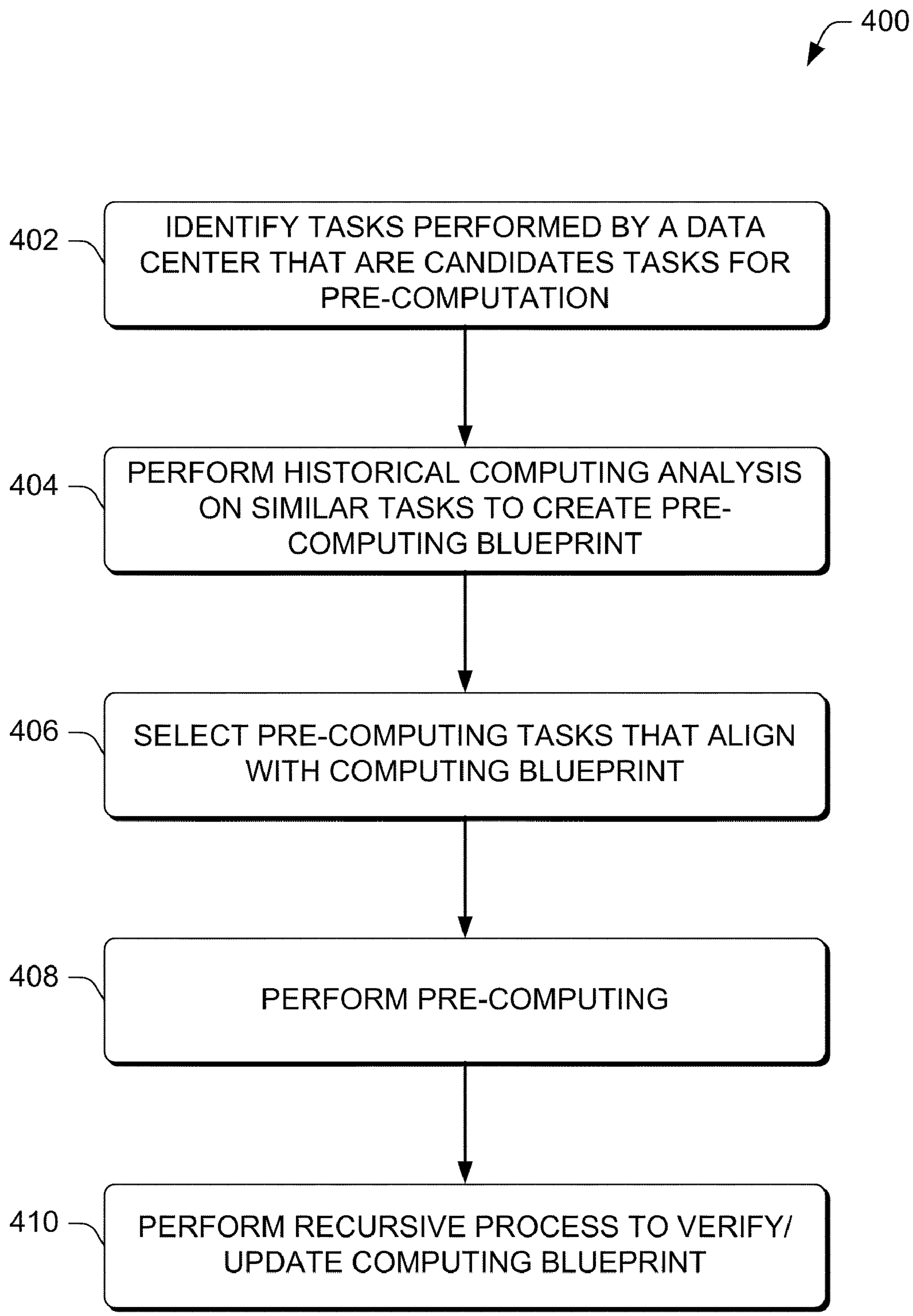

400
402
IDENTIFY TASKS PERFORMED BY A DATA CENTER THAT ARE CANDIDATES TASKS FOR PRE-COMPUTATION
404
PERFORM HISTORICAL COMPUTING ANALYSIS ON SIMILAR TASKS TO CREATE PRE-COMPUTING BLUEPRINT
406
SELECT PRE-COMPUTING TASKS THAT ALIGN WITH COMPUTING BLUEPRINT
408
PERFORM PRE-COMPUTING
410
PERFORM RECURSIVE PROCESS TO VERIFY/ UPDATE COMPUTING BLUEPRINT

FIG. 4

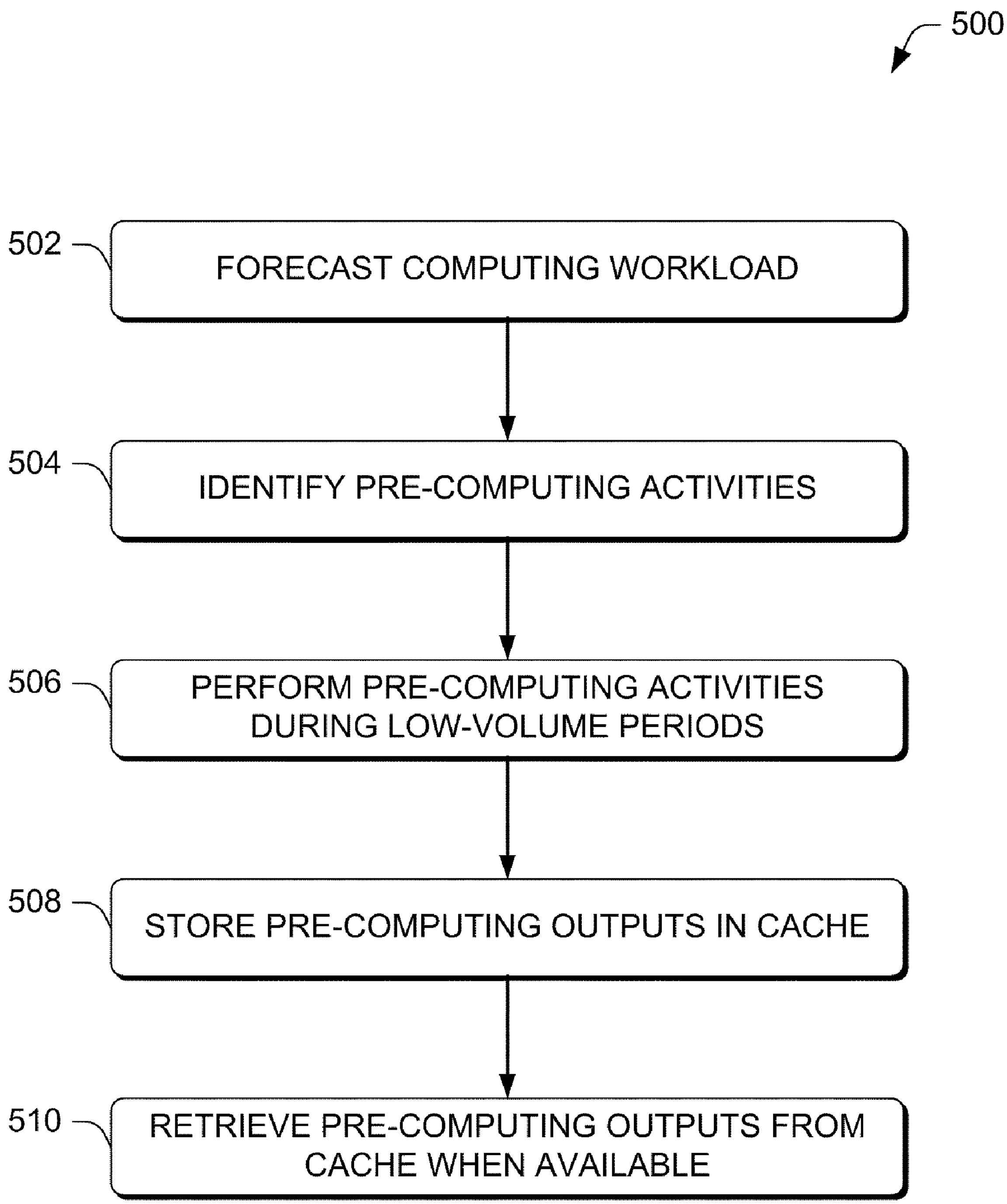

500
502
FORECAST COMPUTING WORKLOAD
504
IDENTIFY PRE-COMPUTING ACTIVITIES
506
PERFORM PRE-COMPUTING ACTIVITIES DURING LOW-VOLUME PERIODS
508
STORE PRE-COMPUTING OUTPUTS IN CACHE
510
RETRIEVE PRE-COMPUTING OUTPUTS FROM CACHE WHEN AVAILABLE

FIG. 5

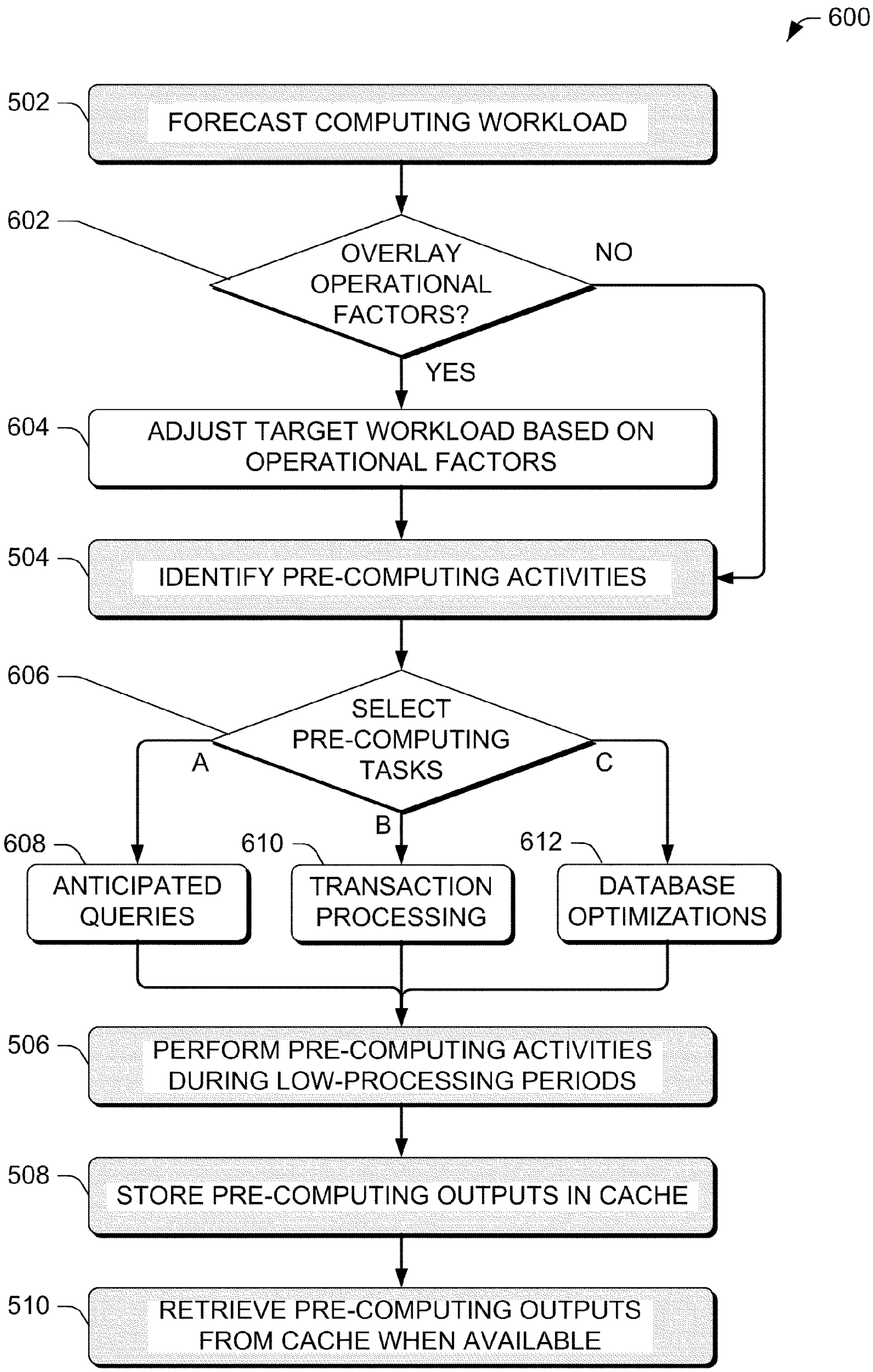

600
502
FORECAST COMPUTING WORKLOAD
602
OVERLAY OPERATIONAL FACTORS?
NO
YES
604
ADJUST TARGET WORKLOAD BASED ON OPERATIONAL FACTORS
504
IDENTIFY PRE-COMPUTING ACTIVITIES
606
SELECT PRE-COMPUTING TASKS
A
B
C
608
ANTICIPATED QUERIES
610
TRANSACTION PROCESSING
612
DATABASE OPTIMIZATIONS
506
PERFORM PRE-COMPUTING ACTIVITIES DURING LOW-PROCESSING PERIODS
508
STORE PRE-COMPUTING OUTPUTS IN CACHE
510
RETRIEVE PRE-COMPUTING OUTPUTS FROM CACHE WHEN AVAILABLE

FIG. 6

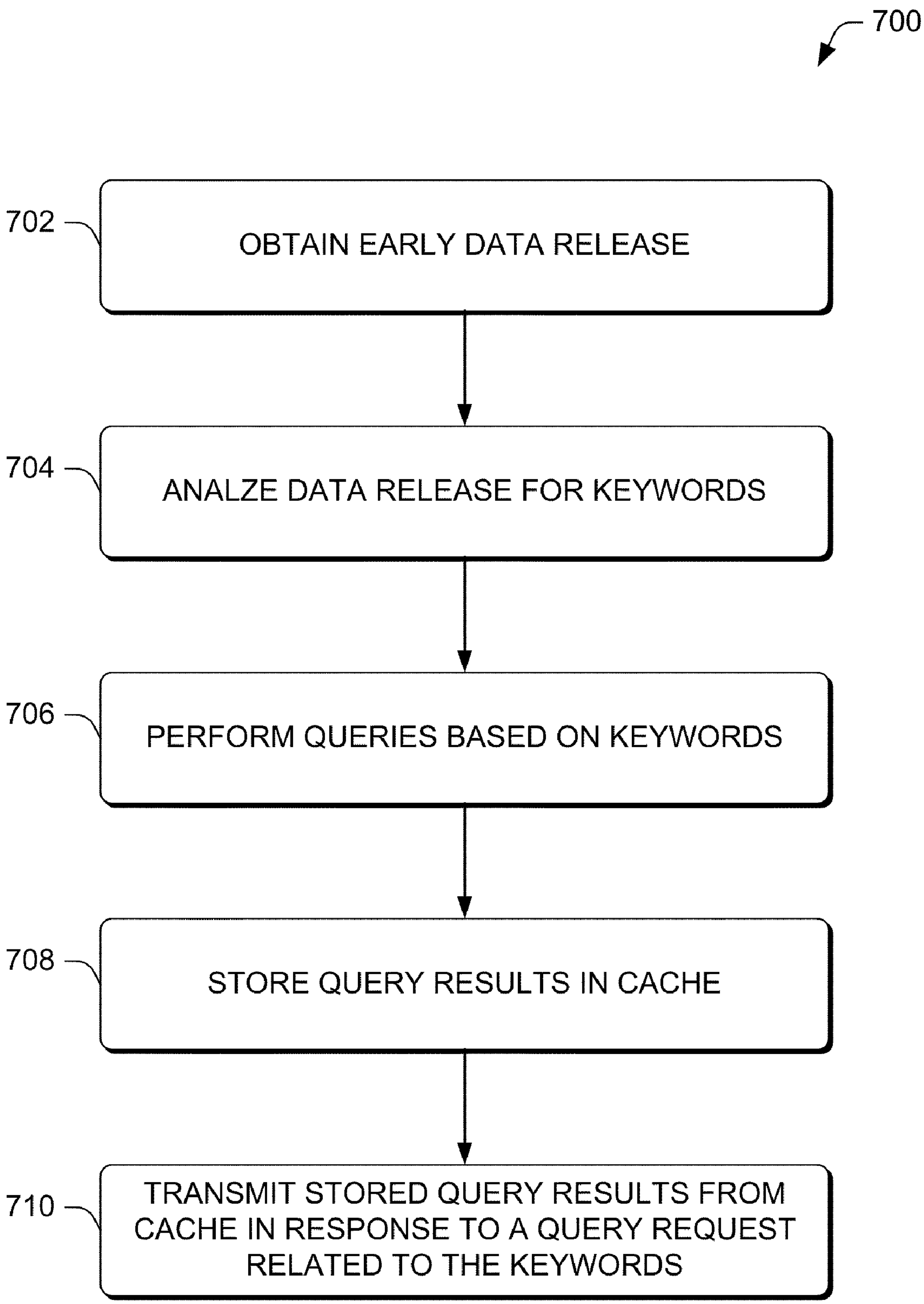

700
702
OBTAIN EARLY DATA RELEASE
704
ANALZE DATA RELEASE FOR KEYWORDS
706
PERFORM QUERIES BASED ON KEYWORDS
708
STORE QUERY RESULTS IN CACHE
710
TRANSMIT STORED QUERY RESULTS FROM CACHE IN RESPONSE TO A QUERY REQUEST RELATED TO THE KEYWORDS

FIG. 7

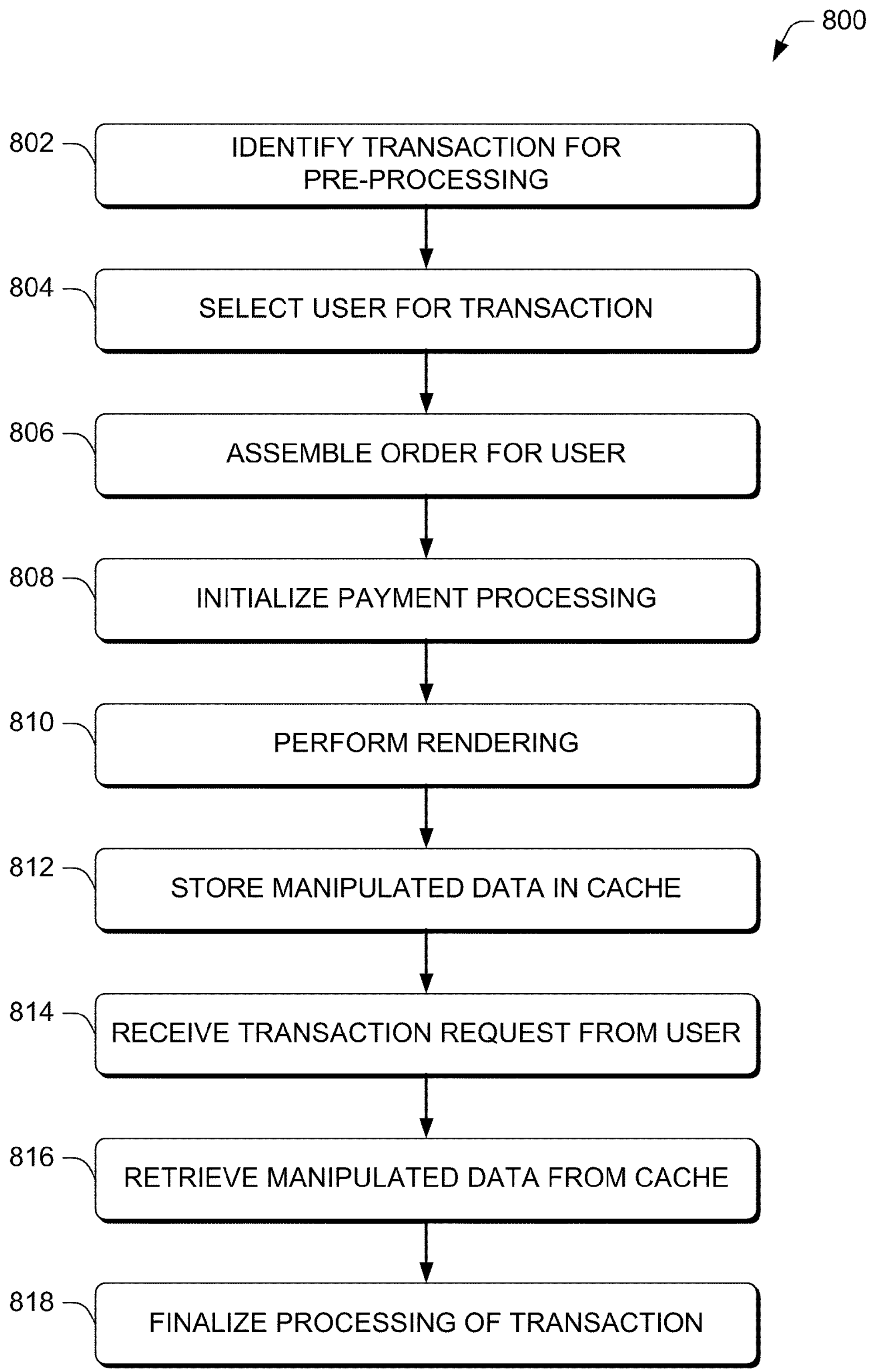

800
802
IDENTIFY TRANSACTION FOR PRE-PROCESSING
804
SELECT USER FOR TRANSACTION
806
ASSEMBLE ORDER FOR USER
808
INITIALIZE PAYMENT PROCESSING
810
PERFORM RENDERING
812
STORE MANIPULATED DATA IN CACHE
814
RECEIVE TRANSACTION REQUEST FROM USER
816
RETRIEVE MANIPULATED DATA FROM CACHE
818
FINALIZE PROCESSING OF TRANSACTION

FIG. 8

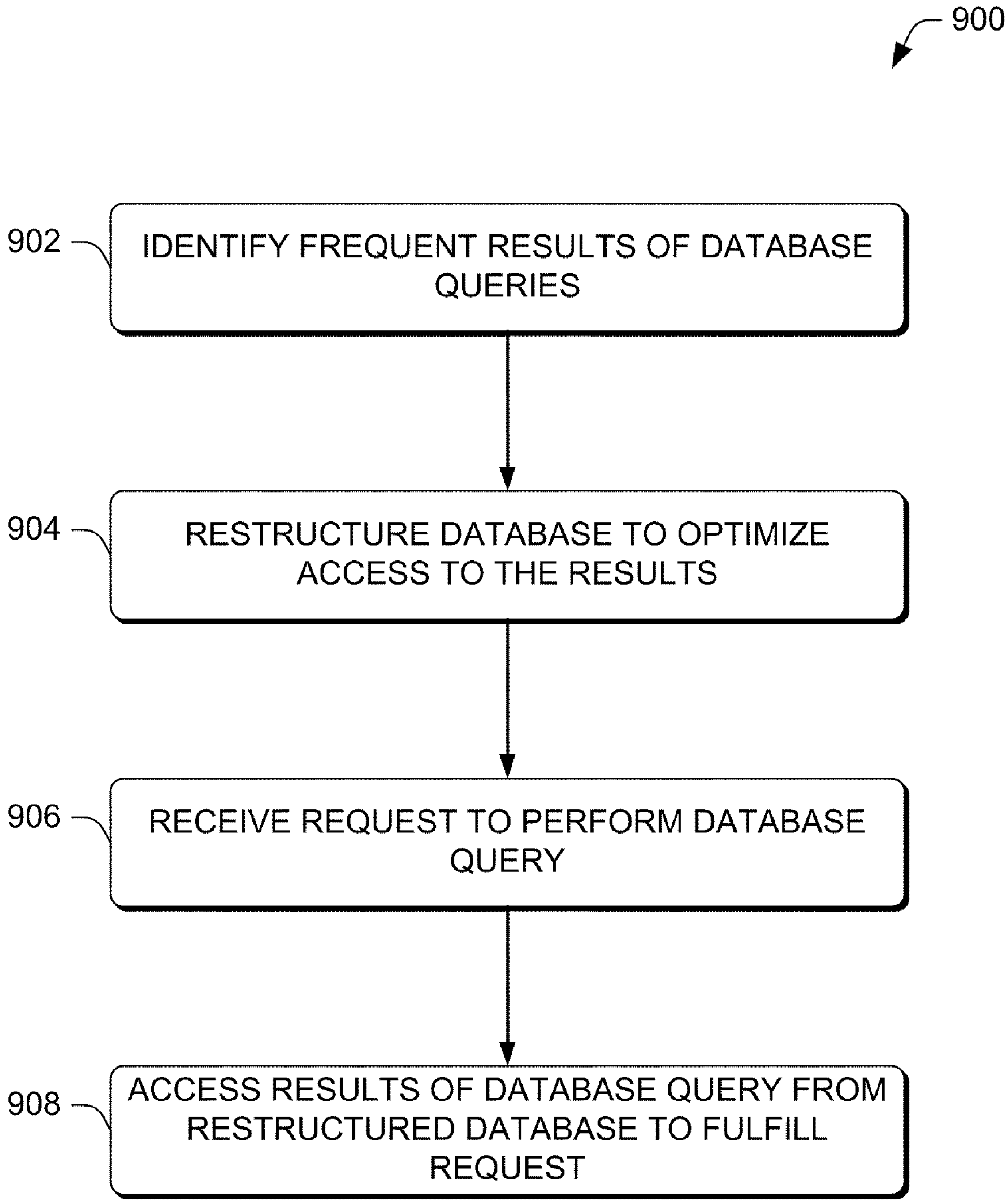

900
902
IDENTIFY FREQUENT RESULTS OF DATABASE QUERIES
904
RESTRUCTURE DATABASE TO OPTIMIZE ACCESS TO THE RESULTS
906
RECEIVE REQUEST TO PERFORM DATABASE QUERY
908
ACCESS RESULTS OF DATABASE QUERY FROM RESTRUCTURED DATABASE TO FULFILL REQUEST

FIG. 9

PRECOMPUTATION FOR DATA CENTER LOAD BALANCING

BACKGROUND

[0001] Data centers often include hundreds to thousands of servers that are configured to process large volumes of data. These data centers may deliver email messages, perform search queries, process retail and bank transactions, stream video and other media, and perform other computation-intensive and high demand computing tasks. Often, a volume of processing at data centers varies greatly over time, which creates periods where the data centers operate near a peak output and other periods where the data centers are underutilized and operate well below the peak output. For example, data centers may experience a lower volume demand late at night or very early in the morning when fewer users are interacting with the data centers.

[0002] Data centers are very expensive to build costing upwards of hundreds of millions of dollars, where the expense relates to a capacity of the data centers. Some data centers are designed with extra capacity that can accommodate a very high computing volume that is experienced during a peak time. However, often the data center operates below capacity and the extra capacity may be unused (idle). In addition, this approach of designing to maximum capacity may be very costly because peaks may be much larger than an average computing workload, and thus a large portion of the investment to create a data center may only be used reach the peak capacity that is infrequent. This may result in millions of dollars in stranded capacity that could have been better allocated to provide an overall lower computational cost.

[0003] A second approach is to design a data center to perform at or near a mean workload level and delay or otherwise underperform during peak processing times. However, this approach may lead to large data latency, stale data, disappointed users, and other disadvantageous consequences. For example, throttling of CPU's may be employed to limit power consumption but also limits throughput, which may have disadvantageous results.

[0004] Operating costs for data centers are also very expensive. An improvement in efficiency of the data centers may enable a substantial reduction in operational costs as compared to computing workload. For example, by making a data center twice as efficient, it may perform operations that previously required two data centers, and thus the improved data center may have a much lower operation cost than an operation cost of two lower efficiency data centers that have a comparable computing capacity.

SUMMARY

[0005] Server workload may be smoothed to reduce or eliminate peaks in workload by pre-computing in a speculative manner one or more aspect of all or a portion of tasks that are predicted to occur during a high-volume period. Pre-computation may reduce capital and operational costs associated with data centers. Computing tasks performed by the data centers may be analyzed to identify computing tasks that are eligible for pre-computing, and may be performed prior to an actual data request from a user or entity.

[0006] In some aspects, the pre-computing tasks may be performed during a low-volume workload period prior to a high-volume workload period to reduce peaks that typically occur in data center workloads that do not utilize pre-computation. The pre-computing tasks may also be performed when operational factors of the servers or data center are optimal, such as when energy prices are least expensive.

[0007] In further aspects, the pre-computed data containing results and or partial solutions may be stored in memory. Upon a receipt of a request for data that has been pre-computed (at least in part), the pre-computed data or partial result may be retrieved from memory. Further processing may be performed on the manipulated data or partial solutions. Finally, the manipulated data may be transmitted to the end user or entity to satisfy the request using less power and to generate a faster response.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The detailed description is described with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same reference numbers in different figures indicate similar or identical items.

[0009] FIG. **1** is a schematic diagram of an illustrative environment, in which embodiments of forecasted pre-computation and load-balancing may be implemented, that includes a data center of servers that process data and exchange data with users and entities.

[0010] FIGS. **2**A, **2**B, and **2**C are charts that depict illustrative data center metrics. FIG. **2**A shows illustrative data center workload versus time, FIG. **2**B shows illustrative cost per unit of inputs for the data center, and FIG. **2**C shows an illustrative overlay of the data from FIGS. **2**A-**2**B.

[0011] FIG. **3** is a chart that depicts illustrative load-balancing of computing tasks from a high-volume period to a prior low-volume period of operation of a data center.

[0012] FIG. **4** is a flow diagram of an illustrative process to decompose a predicted high-volume workload of a data center to pre-calculate a portion of the workload.

[0013] FIG. **5** is a flow diagram of an illustrative process to forecast a computing workload and then balance the workload to reduce peak volume processing.

[0014] FIG. **6** is a flow diagram of another illustrative process to forecast a computing workload and then balance the workload to reduce peak volume processing.

[0015] FIG. **7** is a flow diagram of an illustrative process to pre-calculate queries performed by a data center.

[0016] FIG. **8** is a flow diagram of an illustrative process to pre-compute transactions performed by a data center.

[0017] FIG. **9** is a flow diagram of an illustrative process to optimize databases accessed by a data center to reduce computation time.

DETAILED DESCRIPTION

Overview

[0018] Pre-computing a portion of forecasted workloads may enable load-balancing of data center workload, which may ultimately reduce capital and operational costs associated with data centers. For example, handling the bottlenecks in processing caused by intermittent spikes in computing needs can lead to significant increases in data center capacity—capacity that is rarely used. Computing tasks performed by the data centers may be analyzed to identify computing tasks that are both eligible for pre-computing—and thus may be performed speculatively prior to an actual data request from a user or entity—and that contribute to spikes.

[0019] As an example, a typical data center computation is performed in response to a user request and may include various computation tasks, such as tasks 1 to n: searching a database, compiling data, rendering a result, transmitting data to the user, and so forth. A portion of these example tasks may be performed prior to receipt of the user request by speculatively performing (pre-computing) computations that are likely to be requested by the users or entities (e.g., consumers, researchers, banks, retailers, etc.).

[0020] In some embodiments, the pre-computing may be performed when the data center is experiencing a low-volume workload prior to the high-volume workload that would likely include an anticipated user request. In various embodiments, the speculative workload may be allocated to periods of time that include lower operational cost associated with the computations. In addition, the speculative workload may be allocated using a combination of low-volume and low operational cost considerations.

[0021] The processes and systems described herein may be implemented in a number of ways. Example implementations are provided below with reference to the following figures.

Illustrative Environment

[0022] FIG. **1** is a schematic diagram of an illustrative environment **100** for forecasted pre-computing and load-balancing that includes a data center **102** that processes and exchanges data with other computing devices. The data center **102** may include servers **104** that are capable of performing many different types of computations such as delivering email, streaming data, querying search results, compiling data, processing transactions, and so forth.

[0023] The servers **104** at the data center **102** may perform computations for users **106** that may interact with the data center servers using computing devices **108** (or simply "clients") via one or more networks **110**. A non-exhaustive list of possible clients **108** may include a personal digital assistant, a personal computer, a mobile telephone, a gaming console, an electronic book, and a music player. For example, the users **106** may request search results from the servers **104** at data center **102**, which may, upon receipt of the requests, process computational tasks and transmit a response of search results to the users. In another example, the users **106** may request goods and services by submitting transaction requests to the data center, which, in turn, may process the transactions by performing various computational tasks to carry out the transactions.

[0024] In addition, the servers **104** at data center **102** may perform computations for entities **112** having entity servers **114** in communication with the data center via the networks **110**. For example, the servers **104** at the data center **102** may process transactions (e.g., financial, inventory, etc.) for the entities, maintain databases (storage, backup, etc.) and perform other computing activities for any one of the entities **112**.

[0025] An illustrative processing chart **116** may represent a workload of the servers **104** at the data center **102** as they process computations for the users **106** and/or the entities **112**. The chart plots a workload with respect to time to show peaks and troughs in workload of the servers **104** over time. By implementing pre-computing, as disclosed herein, the data center **102** may operate with minimal fluctuations in workload such that the deviation between peak and a median workload is minimal.

[0026] The data center **102** is in connection to power sources **118**. The power sources **118** may include a variety of sources such as a power plant **118(1)**, solar panels **118(2)**, a hydroelectric power plant **118(3)**, and wind turbines **118**(*m*), among other possible power sources. Each power source may be directly connected to the data center (e.g., dedicated supplies) or in connection with the data center via a shared power grid supply where electricity is purchased for consumption by the data center **102**. The availability and/or price of each source **118(1)**-(*m*) may vary by factors such as availability of sunlight, wind, water, or other resources, or other factors such as time of day, current demand, etc.

[0027] The data center **102** may also include operational factors. For example, in some embodiments, the data center **102** may include a cooling station **120**, which may cool the servers **104** by dissipating heat (e.g., via exposure to water, air, or other elements of a lower temperature, etc.) to enable the servers **104** to operate within predefined operational temperature thresholds. In accordance with embodiments, pre-computing may enable workload to be processed by the servers **104** during optimal operation periods such as when energy is less expensive, cooling costs are minimized, and so forth.

[0028] As illustrated, the data center **102** is equipped with the servers **104** that include a computing infrastructure to perform forecasted pre-computation for load-balancing of data center workload. The servers **104** may include one or more processors **122** and memory **124** that is accessible by the processors. The memory **124** may include applications, modules, and/or data. In some embodiments, the memory **124** may include a scheduler module **126** and a decomposer module **128**.

[0029] In some embodiments, the scheduler module **126** may forecast computing workloads of the servers **104** at the data center **102** to create a forecast of the chart **116**. In this way, the scheduler module **126** may identify high-volume peaks prior to an occurrence of the peaks. Similarly, the forecast may indicate any low-volume troughs prior to the peaks, which may be advantageous for performing extra computing tasks such as pre-computation to load-balance workload of the servers **104** at the data center **102**.

[0030] In various embodiments, the decomposer module **128** may identify computing tasks from an expected high-volume peak, which may be pre-computed prior to the peak. The decomposer module **128** may leverage historical trends to optimize database access, process anticipatory queries, perform various steps of transactions, or perform other tasks prior to an initiation of a request from the users **106** and/or the entities **112** of the computations. The decomposer module **128** may store manipulated data **132** that results from the pre-computations in cache **130**. The servers **104** may access the manipulated data **132** to expedite computations in response to a request from the users **106** and/or the entities **112**, while reducing user-perceived latency as a byproduct of the pre-computation.

[0031] The servers **104** may also include one or more communication adapters **134** to enable communication with the clients **108** and/or the entity server(s) **114**, via the network(s) **110**.

Illustrative Scheduling

[0032] FIGS. **2A**, **2B**, and **2C** are charts that depict illustrative data center operational metrics. The FIGS. **2A-C** reference some of the elements discussed previously in FIG. **1**. FIG. **2A** shows an illustrative chart **200** that depicts data center workload versus time. The chart **200** shows a data center workload **202** plotted over time (t). The magnitude of the workload (w) varies over time. In some embodiments, the scheduler module **126** may forecast the workload **202** to implement pre-computation. The chart **200** also includes a mean workload **204** and a maximum workload indicator **206**, which are reference points plotted on the chart **200** relative to the workload **202**.

[0033] As shown in the illustrative chart **200**, the workload **202** includes high-volume (peaks) portions **208** and low-volume (troughs) portions **210**. The high-volume portions **208** may occur at random times (e.g., following an accident) or at reoccurring times, such as just before 4 pm Eastern Time (ET) when the stock mark is about to close and many investors query current stock prices. The low-volume portions **210** may often occur when most people are sleeping, e.g., during holidays, etc. In another scenario, workload of the data centers **102** would be smooth (constant) and thus follow the mean workload **204**. To accomplish this, computing that occurs at the high-volume portions **208** would be redistributed to the low-volume portions **210** to balance the load of the servers **104** of the data center **102**.

[0034] A margin **211** shows a difference between the mean workload **204** and the maximum workload indicator **206**. The margin **211** represents a stranded cost of overhead (computing power from extra servers, etc.) that is often idle and is primarily used to meet expected peak workloads. Thus, a reallocation of the margin **211** by balancing workload via speculation, as disclosed herein, may enable more effective use (or elimination) of this extra overhead, thus reducing a cost of processing workloads.

[0035] FIG. 2B shows an illustrative chart **212** that depicts cost per unit of inputs for the data center **102** versus time. The chart **212** includes a grid supply curve **214** that depicts variations in the price of energy per unit of inputs. The grid supply curve **214** may include costs for electricity from some or all of the power sources **118** of FIG. **1**. A dedicated supply curve **216** may include energy costs/unit for dedicated power sources that may be exclusively used by the data center **102**, such as the wind turbines **118**(*m*) of a wind harvesting farm and the solar panels **118**(**2**) of a solar harvesting farm. As shown in the chart **212**, the dedicated supply curve **216** may include portions for which the cost/unit is low due to factors such as ample sunlight, wind, etc.

[0036] In some embodiments, an environment curve **218** may depict costs associated with operating the data center **102** that are not directly related to energy supplies. For example, cooling the servers **104** may be more effectively performed when ambient air temperature outside the data center **102** is relatively cool (e.g., less than the maximum operating temperature of the servers **104**, etc.), which may enable operation of the data center at a reduced cost of cooling the servers.

[0037] In accordance with various embodiments, the chart **212** may include a cumulative (operational) curve **220** that combines one or more of the various curves **214**, **216**, and **218** to create a relationship of price per unit associated with operation of the data center **102**. As shown in the illustrative chart, the cumulative curve **220** may include a high portion **222** for which operation of the data center **102** may be most expensive based on the factors included in the cumulative curve **220**.

[0038] FIG. 2C shows an illustrative overlay chart **224** including an overlay of the workload **202** of the chart **200** of FIG. **2**A and an inverse curve **226** of the cumulative curve **220** of the chart **212** of FIG. 2B. The inverse curve **226** may indicate an optimal scheduling for balancing the workload of the data center **102** that may result from load-balancing the workload **202**. The inverse curve **226** may include a low portion **228** that corresponds to the high portion **222** of the cumulative curve for which the price per unit of operation of the data center **102** is high, thus indicating a less desirable time period for scheduling workloads. The inverse curve **226** may have a high portion **230** that is equivalent to the mean workload **204** due to computing resource limitations, which enable minimization of capital investments of the servers **104** and other capital. This may enable the data center **102** to operate at a highly efficient state with minimal idle resources. Stated another way, the inverse cumulative curve **226** may provide a balance between a high utilization of resources (i.e., the servers **104**) while leveraging work during ideal time periods to leverage operations cost savings. A target workload **232** may be derived from aspects of the mean workload **204** and/or the inverse cumulative curve **226**, to create a target (goal) workload of the data center **102** after load balancing by pre-computing workload. In the next section, pre-computing concepts are introduced to explain how to shift the workload **202** to match the target workload **232** to achieve these aforementioned goals.

Illustrative Pre-Computation

[0039] FIG. **3** is a chart **300** that depicts illustrative load-balancing of computing tasks from a high-volume period to a prior low-volume period of operation of the data center **102** of FIG. **1**. The chart **300** includes a workload **302** that depicts illustrative computing workload over time and a mean workload **304**. In some embodiments, the workload **302** may be a forecasted workload that is generated by the scheduler module **126**. The workload **302** includes a high-volume portion **306** and low-volume portions **308** that occur prior to the high-volume portion.

[0040] In accordance with various embodiments, the workload **302** includes millions of computing tasks performed for each time slice **310** at (e.g., at t(x)) by the servers **104** at the data center **102**. The time slice **310** may include fixed tasks **312** which are dependent on user requests, and thus cannot be processed in advance of the user requests. For example, the fixed tasks **312** may include computing requests that deliver a time-sensitive output to the user because the output cannot be adequately performed in advance of a user request.

[0041] In various embodiments, the time slice **310** may also include pre-computable tasks which may be performed in advance of the user requests. For example, pre-computable tasks **314** may include repetitive tasks identified by the decomposer module **128** such as simple searches of high hit-rate subject matter (e.g., celebrity searches, news items, etc.) or queries that may be pre-computed in advance of an arbitrarily scheduled time, and so forth. For example, a feed in a texting feed or other source of evidence that there will be a breaking, popular news story can lead to the proactive generation of news summary pages and search results pages linked to forthcoming expected queries, so as to minimize the expected spikes that will be occurring when awareness and interest grows. Sources of early warning and guidance on future interests by large populations, as well as patterns of evidence that predict the nature and distribution of future interest can be learned with statistical machine learning methods from case libraries over time and then used in similar situations to predict future load and to guide speculative computing aimed at using computation in advance of forthcoming loads to maximally reduce the future real time load on a data center. In the case of a pre-computed and potentially pre-rendered search results, the query search result may be stored in the cache **130** as the manipulated data **132** until a user is identified and the data center **102** can complete the computing by transmitting the manipulated data **132** to a requesting user.

[0042] Such pre-computed results may not be as high of quality as results computed in real time. However, they can be formulated in a manner that makes them partial results, enabling them to be extended and refreshed in real time with lesser amounts of computation than would be required to completely compute and render the search results or other results from scratch in real time.

[0043] Furthermore, such pre-computed results may be used as sufficient fallbacks to reduce real time spikes to be supplanted by freshly computed updates when the computation becomes available given the resources of a data center and target latency tolerances for responsiveness to users.

[0044] The pre-computable tasks **314**, once identified by the decomposer module **128**, may be further identified in a forecast of the high-volume portion **306** and then pre-computed in one of the low-volume portions **308** prior to the occurrence of the high-volume portion **306**, thus load-balancing the workload of the servers **104** at the data center **102**. In accordance with various embodiments, the decomposer module **128** may predict and/or identify types of computing tasks of the servers **104** at the data center **102** that are likely to occur during the high-volume portion **306** (high utilization) and then categorize the computing tasks as the fixed tasks **312** or the pre-computing tasks **314**. In various embodiments, the tasks that occur during the high-volume portion **306** may include a percentage of the pre-computable tasks, that when pre-computed, reduce the anticipated (forecasted) high-volume period to a mean-volume period that has a workload that is equal to or less than the mean curve **304**, or less than the inverse cumulative curve **226** upon consideration of operational factors of the data center.

[0045] FIG. **4** is a flow diagram of an illustrative process **400** to decompose a predicted high-volume workload of the data center **102** to pre-calculate a portion of the workload. The process **400** is described with reference to FIG. **1**, and the decomposer module **128** may provide instructions for some or all of the operations described in the process **400**.

[0046] The process **400** is illustrated as a collection of blocks in a logical flow graph, which represent a sequence of operations that can be implemented in hardware, software, or a combination thereof. In the context of software, the blocks represent computer-executable instructions that, when executed by one or more processors, cause the one or more processors to perform the recited operations. Generally, computer-executable instructions include routines, programs, objects, components, data structures, and the like that perform particular functions or implement particular abstract data types. The order in which the operations are described is not intended to be construed as a limitation, and any number of the described blocks can be combined in any order and/or in parallel to implement the process. Other processes described throughout this disclosure, in addition to process **400**, shall be interpreted accordingly.

[0047] At **402**, the decomposer module **128** may identify tasks that are performed by the data center **102** that are candidates for pre-computation. In some embodiments, high-volume tasks may be strong candidates for pre-computing because of their repetition, ability to be predicted, and cost savings that may result from pre-computation. For example, popular search queries may be identified as high-volume tasks, such as search queries on recent news topics, celebrities, and so forth. In addition, some transactional processes may also be identified as high-volume tasks, such as verifying payments, rendering pages, and so forth.

[0048] At **404**, the decomposer module **128** may perform a historical computing analysis on tasks that are similar to the high-volume tasks (or other tasks identified at the operation **402**) to create a pre-computing blueprint. The blueprint may indicate trends, patterns, repetitive actions, instructions, or other indicators from computing history that help identify tasks for pre-computing which may ultimately help load-balance the workload **202** of the servers **104** at data center **102**. For example, the decomposer module **128** may identify historical trends that indicate a percentage of user searches are predictable and may be pre-computed and stored in cache prior to a user request of the search. In addition, standard search terms may be used in a pre-computing process, and then distributed to users who request similar searches using keywords that overlap with the search terms used in the pre-computing. In another example, transactional orders may be pre-computed based on best selling lists, referrals, or other historical relationships that may predict what orders are likely to occur during a high-volume period of workload **202** of the servers **104** at data center **102**.

[0049] At **406**, the decomposer module **128** may select pre-computing tasks that align with the computing blueprint. The pre-computing tasks may be tagged for pre-computing during an available period, particularly when the servers **104** experience a low-volume workload.

[0050] At **408**, the pre-computing may be performed, and the manipulated data **132** may be stored in the cache **130** for use at a later time (i.e., during the expected high-volume period).

[0051] Finally, at **410**, the decomposer module **128** may perform recursive processes to verify and/or update the computing blueprint from the operation **404** based on actual user computing as compared to the forecasted workload **302** (or workload **202**) at **408**. For example, when pre-computing occurs, the manipulated data **132** may be stored in the cache **130**. Some of the manipulated data **132** may not be used because users might not request this data at a later point in time. Thus, some data may become stale (obsolete), or otherwise not be useful. This data may be disregarded after a predetermined passage of time (expiration). The recursive processes may identify the manipulated data **132** that is used and the manipulated data that is not used to make future decisions on what tasks to pre-compute, thus avoiding pre-computation of tasks that creates unused instances of the manipulated data.

[0052] In accordance with some embodiments, the process **400** may conduct speculative computing (pre-computation) based on statistical models that are generated from analysis of a combination of user history, machine activity, and data center performance/responsiveness that forecast future needs and future bottlenecks. In addition, the predictions can be optimally coupled with simulators that identify the data that is an ideal candidate for pre-computation to minimize potential bottlenecks in real time or near-real time. Thus, the process **400** may seek predictions about the tasks that can be expected to maximally contribute to bottlenecks at data centers and used to guide the speculative computing.

Illustrative Operation

[0053] FIG. **5** is a flow diagram of an illustrative process **500** to forecast a computing workload and then balance the workload to reduce peak volume processing. The process **500** is described with reference to FIGS. **1** and **2**. The scheduler module **126** and the decomposer module **128** may provide instructions for some or all of the operations described in the process **500**.

[0054] At **502**, the scheduler module may forecast a computing workload (i.e., schedule) for the servers **104** of the data center **102**. The forecasted computing workload may be a workload over a given period of time, on a scale of seconds, minutes, hours, or even days. The forecasted workload may include expected high-volume portions, which may be based on historical trends (e.g., close of stock market, early Monday morning business processing, etc.). In some embodiments, the forecasted workload may be performed at near-real time. In near-real time, the scheduler module **126** may detect an upward trend in workload and determine a peak is likely to occur, which may then prompt pre-computing as discussed below.

[0055] At **504**, the decomposer module **128** may determine pre-computing activities that may be performed in advance of a user request for data. In some instances, the decomposer module **128** may determine a new computing task has become repetitively requested, such as a query of an event that recently appeared in the news.

[0056] At **506**, the decomposer module **128** may perform the pre-computing activities during low-volume portions **210**. The low-volume portions **210** may be just prior to the high-volume portions (millisecond to seconds before) or may be performed further in advance of the high-volume portions (minutes, hours, or even days before).

[0057] At **508**, the decomposer module **128** may then store pre-computing results as the manipulated data **132** in the cache **130**.

[0058] At **510**, the decomposer module **128** may retrieve the pre-computing tasks (the manipulated data **132**) from the cache **130** for transmission to users/entities in response to a request. The manipulated data **132** may be further processed after a user request to finalize the request or transmitted directly to a user to fulfill a request. For example, a search query result may that is pre-computed may be transmitted to users that submit similar queries in the future, thus effectively pre-computing workload to reduce future loads (load balance) on the servers **104**. In another example, a portion of a transaction may be pre-computed prior to a user request. When the servers **104** receive the user request of the transaction, any necessary final steps may occur to the manipulated data **132** stored in the cache **130** to finalize the transaction (e.g., authorizing payment on an e-commerce sale, updating inventory, etc.).

[0059] FIG. **6** is a flow diagram of another illustrative process **600** to forecast a computing workload and then balance the workload to reduce peak volume processing. As indicated in FIG. **6**, some of the operations are the same or similar to operations performed in the process **500** and will not be discussed in further detail.

[0060] At **602**, the scheduler module **126** may determine whether to overlay operational factors in the forecast of the workload. The operational factors may include the grid supply curve **214**, the independent supply curve **216**, the environmental curve **218** and so forth, which may be implemented as the cumulative (operational) curve **220** (or the inverse cumulative curve **226**) which are shown in FIGS. **2**B-**2**C.

[0061] At **604**, the scheduler module **126** may adjust the target workload **232** (resulting from load-balancing by pre-computations) based on the operational factors. The target workload **232** may be influenced by the inverse cumulative curve **226**. In accordance with various embodiments, workload may be pre-computed prior to a high-volume period and when the operational factors (i.e., the cumulative curve) indicate that the operation expense of the data center is relatively lower than a mean operational expense.

[0062] At **606**, the decomposer module **128** may select one or more pre-computing tasks.

[0063] At **608** (via route "A"), anticipated queries may be pre-computed. The anticipated queries may be queries that the decomposer module **128** determines are likely to be requested during the high-volume periods, which may be pre-computed and stored in the cache **130** as the manipulated data **132**. For example, search queries on news items or other headlines may be pre-computed queries.

[0064] At **610** (via route "B"), the decomposer module **128** may pre-compute transactions. Some transactions may be identified, via negotiations or other indicators, as capable of being pre-computed. Other transactions, such as retail orders, bank processing, etc., may be speculatively pre-computed, at least in part. In this way, the manipulated data **132** that results from the pre-computation may be used to finalize the transactions or may be discarded if it is not needed.

[0065] At **612** (via route "C"), the decomposer module **128** may optimize databases during low-volume periods to enhance the performance of database operations (queries, etc.) during the high-volume periods. In this way, the reduction in performance may reduce the peaks experienced by the servers **104** at the data center **102** during the high-volume periods.

[0066] The operations **608**, **610**, and **612** are discussed in greater detail below with reference to FIGS. **7**, **8**, and **9**, respectively.

Additional Embodiments

[0067] FIG. **7** is a flow diagram of an illustrative process **700** to pre-calculate queries performed by the servers **104** at the data center **102** of FIG. **1**. Again, the order in which the operations are described is not intended to be construed as a limitation, and any number of the described blocks can be combined in any order and/or in parallel to implement the process. The remaining processes described in this disclosure, in addition to process **700**, shall be interpreted accordingly.

[0068] At **702**, the decomposer module **128** may obtain an early data release prior to a high-volume period of workload of the data center **102**. The early data release may include news headlines after an initial publishing (posting, etc.), or other news that is received in advance of publication or shortly after publication before it has been widely consumed by users.

[0069] At **704**, the decomposer module **128** may analyze the data release for keywords. The keywords may be selected based on the title of an article, the number of uses of the words, the popularity of the word, or other factors. In some instances, the key words may be a person's name or another term of interest. In some embodiments, the blueprint of the operation **404** of FIG. **4** may indicate how to select and/or process the keywords.

[0070] At **706**, the decomposer module **128** may perform queries using the keywords. The queries may be structured based on similar historical queries via the blueprint. For example, when the keywords include a celebrity's name, historical queries that were computed based on celebrity names may be used to structure the query and/or define parameters of the query as the blueprint.

[0071] At **708**, the decomposer module **128** may store the results of the queries from the operation **706** (i.e., the manipulated data **132**) in the cache **130**.

[0072] At **710**, during a high-volume period, the decomposer module **128** may transmit the manipulated data **132** from the cache **130** to the users that submit a search request for related keywords. In some instances, the search queries may not be an exact match of the keyword-based queries that were pre-computed in the operation **706**. However, the manipulated data **132** may be substantially similar such that transmitting the manipulated data **132** to the user will satisfy the user's request while allowing the servers **104** at the data center **102** to perform minimal computations (work) during a high-volume period.

[0073] FIG. **8** is a flow diagram of an illustrative process **800** to pre-compute transactions performed by the servers **104** at the data center **102**.

[0074] At **802**, the decomposer module **128** may identify a transaction for pre-processing. For example, the transaction may be for a sale of goods or services from an e-commerce retailer.

[0075] At **804**, the decomposer module **128** may select a user that is speculatively associated with the transaction. The user may be selected based on various criteria, such as frequency of purchase, likelihood of purchase, and so forth, such that the user is likely to complete the transaction during a high-volume period of operation of the servers **104** at the data center **102**. For example, the transaction may be a release of a new highly anticipated book or movie for which many transactions involving the item are expected to occur following the release.

[0076] At **806**, the decomposing module **128** may assemble the transaction for the user by adding anticipated items (e.g., the book, movie, etc.) to an order.

[0077] At **808**, additional processing may include initializing payment processing, which may include pre-approving funds for the transaction.

[0078] At **810**, the decomposing module **128** may perform rendering of some pages, such as a receipt page, or other pages that may be presented to the user when the user requests the transaction. In some embodiments, more or fewer transactional tasks than those described in the operations **806-810** may be performed by the servers **104** to pre-compute workload of the data center **102**.

[0079] At **812**, the manipulated data **132**, which may include data generated by the operations **804-810**, may be stored in cache.

[0080] At **814**, the servers **104** at the data center **102** may later receive a transaction request from the user, possibly during a high-volume period of workload performed by the servers **104** at the data center **102**.

[0081] At **816**, the servers **104** at the data center **102** may retrieve the manipulated data **132** from the cache **130**.

[0082] At **818**, the processing of the order may be finalized.

[0083] As an example of the process **800**, the transaction may be for a new release of a book that has generated a large demand. The decomposer module **128** may select the users at the operation **804** as users that typically purchase new release titles or related books, such as by querying historical data during a low-volume period (i.e., creating the blueprint). The payment processing may be performed at the operation **808** for users that have their payment information stored in a user account. If the user does not request the transaction at **814**, the manipulated data **132** may be purged from the cache at a later time, such as after the high-volume period or when the manipulated data becomes stale.

[0084] FIG. **9** is a flow diagram of an illustrative process **900** to optimize databases accessed by the data center **102** to reduce computation time.

[0085] At **902**, the decomposer module **128** may identify frequent results of database queries that are often executed by the servers **104**.

[0086] At **904**, the servers **104** may restructure the database to optimize access to the results of the queries. The restructuring may be performed during a low-volume period prior to a high-volume period when the servers **104** at the data center **102** are experiencing a large workload.

[0087] At **906**, the servers **104** at the data center **102** may receive a request to perform the database query that has results that have been optimized at the operation **904**.

[0088] At **908**, the servers **104** at the database **102** may access the results of the database query from the restructured database during the high-volume period, and thus fulfill the request using fewer computing resources than necessary had the database not been restructured for the optimization during the low-volume period of workload of the data center **102**.

[0089] The process **900** may exploit load balancing opportunities by pre-configuration and/or preprocessing of a database to enable the database to provide more efficient access (faster response time, less resource consumption, etc.) in response to real time or near-real time requests. For example, generation of materialized views may be performed during the process **900**, which may make database access more efficient given a probability distribution over anticipated forthcoming queries.

Conclusion

[0090] The above-described techniques pertain to forecasted pre-computation for load balancing of data centers. Although the techniques have been described in language specific to structural features and/or methodological acts, it is to be understood that the appended claims are not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing such techniques.

What is claimed is:

1. A method of load-balancing data center workload, the method comprising:

predicting a high-volume period having a forecasted workload that exceeds a target workload level;

selecting computing tasks that are anticipated to occur during the forecasted workload for pre-computation;

scheduling the pre-computation of the computing tasks during a low-volume period prior to the high-volume period, the low-volume period having a workload that is below a target workload level;

performing the pre-computation of the computing tasks to generate manipulated data; and storing the manipulated data in cache,

retrieving the manipulated data upon a request during the high-volume period to expedite processing of the request that uses the manipulated data.

2. The method as recited in claim **1**, wherein the scheduling includes an overlay of data center operational factors that indicate efficient periods to schedule workloads for pre-computation that are preferentially selected over non-efficient periods.

3. The method as recited in claim **2**, wherein the data center operational factors include at least one of a cost, power consumption, or an availability of an energy source.

4. The method as recited in claim **1**, wherein the pre-computation includes at least one of:

pre-computing anticipated queries;

pre-computing anticipated transactions; and

optimizing databases to enable expedited results of database queries.

5. The method as recited in claim **1**, wherein the selecting computing tasks is based on a historical analysis of tasks that are anticipated to reoccur in the future.

6. The method as recited in claim **1**, wherein the selecting occurs at near-real time as a data center workload enters the high-volume period.

7. A data center processing system, comprising:

a communication adapter to enable a data center to receive a request from an entity computing device and transmit a response to the entity computing device;

one or more processors; and

memory to store computer readable instructions executable by the processor, the memory used to store a decomposer module to:

identify computing tasks to be pre-computed in anticipation of the request expected to occur during a high-volume period of workload of the data center, the computing tasks to fulfill at least a portion of the request;

pre-compute the computing tasks during a low-volume period to create manipulated data based on a computing blueprint that uses historical computing trends to determine how to pre-compute the computing tasks; and

store the manipulated data in cache for access during the high-volume period.

8. The system as recited in claim **7**, wherein the decomposer module is to further:

retrieve a portion of the manipulated data to expedite satisfying the request upon receipt of the request; and

delete unused instances of the manipulated data when the manipulated data becomes stale.

9. The system as recited in claim **7**, wherein the computing tasks include queries that are pre-computed by selecting keywords from an early data release.

10. The system as recited in claim **7**, wherein the computing tasks include transactions that are pre-computed by inputting at least one of user information, item information, and payment information.

11. The system as recited in claim **7**, wherein the computing tasks include an optimization of databases to enable expedited results of database queries.

12. The system as recited in claim **7**, wherein the memory is to further store a scheduler module to:

predict workload volume of the data center; and

schedule the pre-compute to occur during a low-volume workload prior to a high-volume workload of the data center when the computing tasks are requested.

13. The system as recited in claim **12**, wherein the scheduler module is to further include an overlay of data center operational factors that indicate optimum workload scheduling to obtain an efficient use of data center resources.

14. A method of processing a portion of a computing workload of a data center prior to a request of the workload, the method comprising:

identifying computing tasks performed by the data center to be pre-computed prior to an initiated request;

obtaining inputs for pre-computing based on a historically based blueprint that uses historical trends to predict future computing demands;

performing the pre-computation of computing tasks during a low-volume workload period of the data center prior to a high-volume workload period, the pre-computing tasks to create manipulated data; and

storing the manipulated data in cache.

15. The method as recited in claim **14**, further comprising retrieving the manipulated data in response to a received request that requires the computing tasks.

16. The method as recited in claim **14**, wherein the inputs are keywords selected form an early data release, the keywords used as search terms for queries that are pre-computed to create the manipulated data.

17. The method as recited in claim **14**, wherein the inputs include transaction information to enable pre-computing of a transaction by performing at least one of compiling a portion of a transaction, adding items to an order, processing a payment, or assigning a customer.

18. The method as recited in claim **14**, further comprising generating a forecasted schedule of workload of the data center that includes the low-volume period and the high-volume period.

19. The method as recited in claim **18**, wherein the forecasted schedule includes an overlay of data center operational factors that indicate optimum workload scheduling to obtain an efficient use of data center resources.

20. The method as recited in claim **14**, wherein identifying computing tasks performed by the data center includes speculative computing based on statistical models using at least one of user history, machine usage, or data center performance, the statistical models coupled with simulators to enable identification of data for pre-computation.

* * * * *